

Implicit Adaptive Mesh Refinement for 2D Reduced Resistive Magnetohydrodynamics

Bobby Philip, Luis Chacón

Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545

Michael Pernice

Center for Advanced Modeling and Simulation, Idaho National Laboratory, Idaho Falls, ID 83415-2211

Abstract

An implicit structured-adaptive-mesh-refinement (SAMR) solver for 2D reduced magnetohydrodynamics (MHD) is described. The time-implicit discretization is able to step over fast normal modes, while the spatial adaptivity resolves thin, dynamically evolving features. A Jacobian-free Newton-Krylov method is used for the nonlinear solver engine. For preconditioning, we have extended the optimal “physics-based” approach developed in [11] (which employed multigrid solver technology in the preconditioner for scalability) to SAMR grids using the well-known Fast Adaptive Composite grid (FAC) method [35]. A grid convergence study demonstrates that the solver performance is independent of the number of grid levels and only depends of the finest resolution considered, and that it scales well with grid refinement. The study of error generation and propagation in our SAMR implementation demonstrates that high-order (cubic) interpolation during regridding, combined with a robustly damping second-order temporal scheme such as BDF2, is required to minimize impact of grid errors at coarse fine interfaces on the overall error of the computation for this MHD application. We also demonstrate that our implementation features the desired property that the overall numerical error level is dependent only on the finest resolution level considered, and not on the base-grid resolution or on the number of refinement levels present during the simulation. We demonstrate the effectiveness of the tool on several challenging problems.

Key words: Adaptive mesh refinement, Newton-Krylov, implicit methods, magnetohydrodynamics, multilevel solvers

PACS:

Email addresses: `bphilip@lanl.gov`, `chacon@lanl.gov` (Bobby Philip, Luis Chacón), `michael.pernice@inl.gov` (Michael Pernice).

1 Introduction

The magnetohydrodynamics (MHD) model is useful for studying the macroscopic behavior of fully ionized gases (plasmas). Plasmas exhibit a wide range of complex behavior, and are intrinsically multiscale both temporally and spatially. While MHD provides a tractable model for the macroscopic description of plasmas, it still presents formidable challenges for the numerical modeler. In particular, MHD (even in its simplest form) supports multiple time scales (which manifest in the form of waves) and multiple length scales (which manifest in the form of microscopic layer formation, often with macroscopic relevance).

Algorithmically, the multiscale nature of MHD needs to be addressed separately in time and space. Spatially, the dynamic formation of thin layers requires grid adaptation that can respond dynamically. While there are many options available for dynamic grid adaptation depending on the spatial representation of choice (e.g., r -refinement, h -refinement, p -refinement), our focus here is on h -refinement in the finite-volume context via structured adaptive mesh refinement (SAMR). A SAMR grid is organized as a hierarchy of nested refinement levels, with each level comprised of a union of rectangular patches. As the locally refined grid evolves to follow important features in the solution, these levels are created and destroyed as needed, and the solution is transferred from the old grid to the new grid to continue the simulation. We use SAMRAI [26] to handle this complexity of grid and data management. SAMR benefits from a structured block-based approach, which allows one to straightforwardly retrofit structured-mesh codes into a SAMR framework.

Temporally, our interest is on applications where fast time scales are parasitic to a slower dynamical time scale of interest (such applications arise, for instance, in fusion [48,46] and space [23,42] plasmas). Accordingly, it is of interest to step over such fast time scales in order to resolve those of dynamical interest, while preserving the temporal accuracy of the approach. Explicit time integration methods are subject to stability constraints that arise from the fastest time scales, and are inappropriate for this purpose because they force the modeler to follow the fastest time scale supported. Fully-implicit time integration methods allow stepping over fast time scales, since time steps are generally constrained only by accuracy, not stability. However, they require the solution of large-scale systems of nonlinear equations at each time step, and fast, robust solution methods are necessary for implicit methods to be practical. Fortunately, Newton-Krylov methods [8] have provided such robust solvers for a variety of contexts [31], including MHD [11,10,9], provided effective preconditioning is used. In Refs. [11,10,9], the key for algorithmic performance was the use of multigrid methods in the preconditioner stage.

Patch-based refinement in the context of MHD has been explored by many previous studies in the literature, both in the context of finite volumes (see e.g. [4,29,48,47,54,22]) and finite (and spectral) elements (e.g., [51,33,44]). In the finite-volume context, these studies have focused on various aspects of both the temporal and spatial discretization of the MHD equations on AMR grids. Spatially, authors have explored both staggered [4] and cell-centered [29,48,47] representations, with special emphasis on the preservation of conserved quantities and the solenoidal property of the magnetic field. An interesting study comparing the accuracy of finite volumes/differences vs. spectral elements in an MHD-AMR context can be found in Ref. [38]. Temporally, most AMR implementations have relied on explicit methods, albeit with some flavor of time-step subcycling for better performance (see e.g. [4,29]). However, a number of authors have explored more advanced time-stepping algorithms, such as partially implicit [47] (where hyperbolic terms are treated explicitly, and diffusive terms implicitly), implicit/explicit [54] (where some blocks are treated explicitly, while others are treated linearly implicitly), and fully implicit [22] (although using inefficient direct solvers).

The focus of this study is to merge the SAMR dynamic adaptive grid approach with efficient, scalable fully implicit time integration, in the context of MHD. For simplicity, we focus our attention on the 2D reduced resistive MHD model [52,15,25], which is rigorously valid in the presence of a large guide magnetic field. The reduced resistive MHD model has the advantage of simplicity while maintaining a truly multiscale character, both temporally and spatially. Furthermore, mature fully-implicit technology is available [11], which will be reused for this study.

The advantages of fully implicit SAMR are obvious, as it enables dynamic refinement while decoupling the time integrator from the small explicit Courant-Friedrichs-Lewy (CFL) stability limits Δt_{CFL} (which scales with the mesh size) that would arise in the patches of finest resolution. Key to the effectiveness and scalability of the proposed approach is to generalize the multigrid treatment proposed in Refs. [11,10] to SAMR grids. This can be achieved with fast multi-level methods that exploit the structure of the mesh, such as the Fast Adaptive Composite grid (FAC) method [35], as has been already demonstrated in the context of 2D radiation-diffusion [40].

Preliminary results on combining implicit time integration with SAMR for resistive MHD were first reported in [41]. Here we expand the study to include considerations of accuracy and provide details of our treatment of discretization at coarse-fine interfaces. Section 2 describes the mathematical model and its numerical discretization. Section 3 introduces the nonlinear solver of choice, Jacobian-free Newton-Krylov methods, and the preconditioning approach to make it efficient. The specifics of the coarse-fine interface treatment for this application are provided in Sec. 4. Finally, numerical results focusing on per-

formance and accuracy aspects of the solver are presented in Sec. 5, and we conclude in Sec. 6.

2 Numerical model: Current-Vorticity Formulation of Reduced MHD

In the 2D reduced MHD (RMHD) formalism, the magnetic field component in the ignorable direction B_z is much larger than the magnitude of the poloidal magnetic field \vec{B}_p . As a result, $B_z \approx \text{constant}$ and the poloidal velocity \vec{v} is incompressible ($\nabla \cdot \vec{v} = 0$), and the general MHD formalism reduces to [52,15,25]:

$$\nabla^2 \Phi = \omega, \quad (1)$$

$$(\partial_t + \vec{v} \cdot \nabla - \frac{\eta}{\mu_0} \nabla^2) \Psi + E_0 = 0, \quad (2)$$

$$\rho(\partial_t + \vec{v} \cdot \nabla - \nu \nabla^2) \omega = \frac{1}{\mu_0} \vec{B} \cdot \nabla J, \quad (3)$$

where Φ is the poloidal velocity stream function ($\vec{v} = \vec{z} \times \nabla \Phi$), ω is the vorticity in the poloidal plane ($\omega = \vec{z} \cdot \nabla \times \vec{v}$), Ψ is the poloidal flux function (which gives $\vec{B}_p = \vec{z} \times \nabla \Psi$), $\vec{B} = \vec{B}_p + B_z \vec{z}$ is the total magnetic field, $J = \nabla^2 \Psi$ is the current, and ρ is the density (which is taken as constant). Note that \vec{B} always satisfies the solenoidal property. The source E_0 (the applied electric field in the z -direction) has been included to balance the resistive decay of the equilibrium. The transport parameters (the kinematic viscosity ν and the resistivity η) are assumed constant. We note that $\vec{B} \cdot \nabla = \vec{B}_p \cdot \nabla$ since $\partial_z = 0$, but we keep \vec{B} for the sake of generality.

Equations 1-3 are normalized as follows: \vec{B} is normalized to the characteristic poloidal magnetic field B_0 , ρ to the constant density ρ_0 , lengths to the domain length in the y -direction L_y , and the time to the poloidal Alfvén time $\tau_A = L_y/v_A$, where $v_A = B_0/\sqrt{\rho_0 \mu_0}$ is the Alfvén speed. The normalized set of RMHD equations reads:

$$\begin{aligned} \partial_t \omega + \vec{v} \cdot \nabla \omega - \nu \nabla^2 \omega &= \vec{B} \cdot \nabla J \\ \partial_t \Psi + \vec{v} \cdot \nabla \Psi - \eta \nabla^2 \Psi &= -E_0 \\ \nabla^2 \Phi &= \omega \end{aligned} \quad (4)$$

where η is the normalized resistivity (the inverse of the Lundquist number) and ν is the normalized viscosity (the inverse of the Reynolds number).

While the form in (4) was successfully treated in [11], we have found that this formulation is not well-suited for SAMR, due to difficulties discretizing the

high-order term $\vec{B} \cdot \nabla J = \vec{B} \cdot \nabla \nabla^2 \Psi$ at coarse-fine interfaces. This is similar to difficulties reported in [51] for an ideal reduced MHD formulation. Instead, following Refs. [51,33], we use the current-vorticity formulation obtained by applying a ∇^2 to the poloidal flux equation above, to obtain:

$$\begin{aligned} \partial_t \omega + \vec{v} \cdot \nabla \omega - \nu \nabla^2 \omega &= \vec{B} \cdot \nabla J \\ \partial_t J + \vec{v} \cdot \nabla J - \vec{B} \cdot \nabla \omega - \{\phi, \psi\} - \eta \nabla^2 J &= 0 \\ \nabla^2 \phi &= \omega \\ \nabla^2 \psi &= J \end{aligned} \tag{5}$$

where $\{\phi, \psi\} = 2[\phi_{xy}(\psi_{xx} - \psi_{yy}) - \psi_{xy}(\phi_{xx} - \phi_{yy})]$. This formulation is advantageous because it avoids derivatives that are higher than second order, and all dependent variables are determined from integration rather than differentiation. However, it features two elliptic constraints instead of one in (4). Furthermore, its implementation requires modifications to the semi-implicit preconditioner developed in [11] for (4), which will be discussed later in this paper (Sec. 3.2).

For the temporal discretization of (5), we will explore two approaches: a θ -scheme ($\theta = 0.5$ is second-order accurate, and corresponds to the Crank-Nicolson scheme [12]) and second-order backward differentiation formula (BDF2) [13,21]. The θ -scheme reads:

$$\begin{aligned} \frac{(J^{n+1} - J^n)}{\Delta t} + [\vec{v} \cdot \nabla J]^{n+\theta} - \eta \Delta J^{n+\theta} &= [\vec{B} \cdot \nabla \omega]^{n+\theta} + \{\Phi, \Psi\}^{n+\theta} \\ \frac{(\omega^{n+1} - \omega^n)}{\Delta t} + [\vec{v} \cdot \nabla \omega]^{n+\theta} - \nu \Delta \omega^{n+\theta} &= [\vec{B} \cdot \nabla J]^{n+\theta} \\ \Delta \Phi^{n+\theta} &= \omega^{n+\theta} \\ \Delta \Psi^{n+\theta} &= J^{n+\theta} \end{aligned} \tag{6}$$

where $n + \theta$ quantities are calculated as $\xi^{n+\theta} = (1 - \theta)\xi^n + \theta\xi^{n+1}$. The BDF2 scheme discretizes the temporal derivative terms by fitting a quadratic polynomial using the $n + 1$, n , and $n - 1$ time levels, and then differencing it at the new time level $n + 1$. Unlike Crank-Nicolson, BDF2 features robust damping of dissipative terms [21]. The purpose of considering these two approaches is to compare their error propagation properties in the presence of coarse-fine interfaces (Sec. 5).

Within AMR patches, spatial operators in (6) are discretized using second-order centered finite differences. Following [11], advective terms are discretized in non-conservative form, using centered differences. Boundary conditions (either at physical boundaries or at coarse-fine interfaces) are imposed using ghost cells. The spatial treatment of coarse-fine interfaces employed in this application is described in detail later in this paper (Sec. 4).

3 Nonlinear solution algorithm

Our general approach to the solution of (6) is via preconditioned Jacobian-free Newton-Krylov methods (JFNK). These methods have demonstrated their effectiveness in many similar applications [31], including 2D reduced resistive MHD [11] and Hall MHD [10], and 3D resistive MHD [9]. Our approach generalizes that of [11] in two fundamental ways: firstly, we have adapted the preconditioning strategy to deal with the $J - \omega$ formulation instead of the $\Psi - \omega$ formulation; and secondly, we have generalized the single-mesh multi-grid treatment advocated in that reference to AMR meshes using the Fast Adaptive Composite grid (FAC) method [35]. In what follows, we summarize the JFNK philosophy and our approach to preconditioning. The next section will deal with the AMR aspects of this application.

3.1 Jacobian-free Newton-Krylov Methods

Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a nonlinear function and consider calculating the solution $x^* \in \mathbb{R}^n$ of the system of nonlinear equations

$$F(x^*) = 0. \quad (7)$$

Classical Newton's method for solving (7) generates a sequence of approximations x_k to x^* , where $x_{k+1} = x_k + s_k$ and the Newton step s_k is the solution to the system of linear equations

$$F'(x_k)s_k = -F(x_k), \quad (8)$$

where F' is the Jacobian of F evaluated at x_k . Newton's method is attractive because of its fast local convergence properties, but for large-scale problems, it is impractical to solve (8) with a direct method. Furthermore, it is often useless to solve (8) using a tight convergence tolerance when x_k is far from x^* , since the linearization that leads to (8) may be a poor approximation to $F(x)$. Generally, it is much more efficient to employ so-called inexact Newton methods [14], in which the linear tolerance for (8) is selected adaptively by requiring that s_k only satisfy:

$$\|F(x_k) + F'(x_k)s_k\| \leq \eta_k \|F(x_k)\| \quad (9)$$

for some $\eta_k \in (0, 1)$ [14]. When the forcing term η_k is chosen appropriately, superlinear and even quadratic convergence of the iteration can be achieved [16].

While any iterative method can be used to find an s_k that satisfies (9), Krylov subspace methods are distinguished by the fact that they require only matrix-vector products to proceed. These matrix-vector products can be approximated by a finite-difference version of the directional (Gâteaux) derivative as:

$$F'(x_k)v \approx \frac{F(x_k + \varepsilon v) - F(x_k)}{\varepsilon}, \quad (10)$$

which is especially advantageous when F' is difficult to compute or expensive to store (as is the case in this application due to the presence of multiple grid patches). While the selection of a suitable differencing parameter ε may be non-trivial for some applications, it is generally well-understood [28]. For this application,

$$\varepsilon = \sqrt{\epsilon_{\text{mach}}} \frac{\sqrt{1 + \|x_k\|}}{\|v\|},$$

where ϵ_{mach} is machine precision.

Among the various Krylov methods available, GMRES is selected because it guarantees convergence with nonsymmetric, nonpositive definite systems [45] (the case here because of flow and wave propagation), and because it provides normalized Krylov vectors $\|v\| = 1$, thus bounding the error introduced in the difference approximation of (10) (whose leading error term is proportional to $\varepsilon\|v\|^2$) [37]. However, GMRES can be memory intensive (storage increases linearly with the number of GMRES iterations per Jacobian solve) and expensive (computational complexity of GMRES increases with the square of the number of GMRES iterations per Jacobian solve). Restarted GMRES can in principle deal with these limitations; however, it lacks a theory of convergence, and stalling is frequently observed in real applications [32]. Here, we focus on minimizing the number of GMRES iterations per Jacobian solve for efficiency and robustness by: 1) using inexact Newton techniques (as described above), and 2) improving the condition number of the Jacobian matrix by preconditioning the problem. The next section describes our approach to preconditioning.

3.2 Preconditioning

Implicit time differencing eliminates CFL-based stability constraints, allowing us to select time steps independent of the level of mesh refinement, by introducing dispersion in waves and by treating elliptic operators (such as diffusion) nonlocally. However, some of the mechanisms that are sources of

numerical instabilities in explicit methods continue to manifest themselves in implicit schemes in the form of ill-conditioned algebraic systems, which iterative techniques have difficulty in handling.

As is explained in [11], there are two sources of ill-conditioning in the system of MHD equations: elliptic operators and hyperbolic couplings. The former can be dealt with effectively with multilevel techniques. The latter, however, cannot be unless the hyperbolic couplings are reformulated in a multilevel-friendly fashion. References [11,10] provide a systematic way of doing this, which we follow here.

3.2.1 Approximate formulation of the reduced MHD system

Krylov techniques are employed here to approximately solve (8) to the dynamically selected tolerance (9) in each Newton step. Hence, the construction of the physics-based preconditioner necessarily starts from the linearized system of equations. For the system in (5), the linearized equations read (in block form):

$$\begin{pmatrix} \mathcal{L}_\eta & -\theta \mathbf{B}_0 \cdot \nabla & U_{J,\psi} & U_{J,\phi} \\ -\theta \mathbf{B}_0 \cdot \nabla & \mathcal{L}_\nu & U_{\omega,\psi} & U_{\omega,\phi} \\ \mathbb{I} & 0 & -\Delta & 0 \\ 0 & \mathbb{I} & 0 & -\Delta \end{pmatrix} \begin{pmatrix} \delta J \\ \delta \omega \\ \delta \Psi \\ \delta \Phi \end{pmatrix} = \begin{pmatrix} r_J \\ r_\omega \\ r_\psi \\ r_\phi \end{pmatrix} \quad (11)$$

where the diagonal blocks \mathcal{L}_η and \mathcal{L}_ν read

$$\mathcal{L}_\eta = \frac{\mathbb{I}}{\Delta t} + \theta(\mathbf{u}_0 \cdot \nabla - \eta \Delta), \quad \mathcal{L}_\nu = \frac{\mathbb{I}}{\Delta t} + \theta(\mathbf{u}_0 \cdot \nabla - \nu \Delta).$$

and the off-diagonal entries $U_{J,\psi}$, $U_{J,\phi}$, $U_{\omega,\psi}$ and $U_{\omega,\phi}$ are given by:

$$U_{J,\psi} = -\theta(\nabla \omega_0 \cdot \vec{z} \times \nabla + \{\phi_0, \cdot\}), \quad U_{J,\phi} = \theta(\nabla J_0 \cdot \vec{z} \times \nabla - \{\cdot, \psi_0\}),$$

and

$$U_{\omega,\psi} = -\theta \nabla J_0 \cdot \vec{z} \times \nabla, \quad U_{\omega,\phi} = \theta \nabla \omega_0 \cdot \vec{z} \times \nabla.$$

In order to formulate an approximate, multilevel-friendly form of the linearized set, we follow [11] to reduce the order of the δJ , $\delta \omega$ equations by factoring out a Laplacian operator (thus rendering equations for $\delta \Psi$, $\delta \Phi$, respectively). This is done by first eliminating δJ and $\delta \omega$ from the corresponding equations in favor

of $\delta\Psi$ and $\delta\Phi$ (using the linearized elliptic constraints). In the δJ equation, one can factor out the Laplacian operator trivially (since the Laplacian is a linear operator, and the J equation was obtained by applying a Laplacian operator onto the Ψ equation in the first place). In the $\delta\omega$ equation, the Laplacian operator can be factored out approximately in the same fashion as was shown in [11]. After these transformations, there results the approximate system:

$$\mathcal{P} \begin{pmatrix} \delta\Psi \\ \delta\Phi \end{pmatrix} \approx \mathbf{\Delta}^{-1} \left[\begin{pmatrix} r_J \\ r_\omega \end{pmatrix} - \mathcal{P} \begin{pmatrix} r_\Psi \\ r_\Phi \end{pmatrix} \right],$$

where

$$\mathcal{P} \equiv \begin{pmatrix} \mathcal{L}_\eta & -\theta \mathbf{B}_0 \cdot \nabla \\ -\theta \mathbf{B}_0 \cdot \nabla & \mathcal{L}_\nu \end{pmatrix}$$

is the same hyperbolic operator found in [11]. After solving for $\delta\Psi$ and $\delta\Phi$, one can recover δJ and $\delta\omega$ by solving:

$$\mathcal{P} \begin{pmatrix} \delta J \\ \delta\omega \end{pmatrix} = \begin{pmatrix} r_J - \theta(\delta \mathbf{u} \cdot \nabla J_0 - \delta \mathbf{B} \cdot \nabla \omega_0 - \{\delta\Phi, \Psi_0\} - \{\Phi_0, \delta\Psi\}) \\ r_\omega - \theta(\delta \mathbf{u} \cdot \nabla \omega_0 - \delta \mathbf{B} \cdot \nabla J_0) \end{pmatrix},$$

which again requires inverting \mathcal{P} .

Following [11], systems of equations $\mathcal{P}\mathbf{v} = \mathbf{b}$ are solved with a few iterations of the stationary method obtained from the splitting

$$\mathcal{P} = \underbrace{\begin{pmatrix} \mathcal{L}_\eta & -\theta \mathbf{B}_0 \cdot \nabla \\ -\theta \mathbf{B}_0 \cdot \nabla & \mathcal{D}_\nu \end{pmatrix}}_{\mathcal{M}} - \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{D}_\nu - \mathcal{L}_\nu \end{pmatrix},$$

with \mathcal{D}_ν the diagonal of the advection diffusion operator \mathcal{L}_ν . This splitting results in the iteration:

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \mathcal{M}^{-1}(\mathbf{b} - \mathcal{P}\mathbf{v}^k).$$

The inversion of \mathcal{M} involves first a block factorization:

$$\mathcal{M} = \begin{pmatrix} \mathbb{I} & -\theta(\mathbf{B}_0 \cdot \nabla) \mathcal{D}_\nu^{-1} \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} P_{SI} & 0 \\ 0 & \mathcal{D}_\nu \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ \theta \mathcal{D}_\nu^{-1}(\mathbf{B}_0 \cdot \nabla) & \mathbb{I} \end{pmatrix},$$

with $P_{SI} = \mathcal{L}_\eta - \theta^2 \nabla \cdot (\mathbf{B}_0 \mathcal{D}_\nu^{-1} \mathbf{B}_0^\top \nabla)$, and then the inversion of the resulting matrices, yielding:

$$\mathcal{M}^{-1} = \begin{pmatrix} \mathbb{I} & 0 \\ -\theta \mathcal{D}_\nu^{-1} (\mathbf{B}_0 \cdot \nabla) & \mathbb{I} \end{pmatrix} \begin{pmatrix} P_{SI}^{-1} & 0 \\ 0 & \mathcal{D}_\nu^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & \theta (\mathbf{B}_0 \cdot \nabla) \mathcal{D}_\nu^{-1} \\ 0 & \mathbb{I} \end{pmatrix}.$$

The implementation of \mathcal{M}^{-1} only requires the (trivial) inversion of \mathcal{D}_ν (which is a diagonal matrix), and the inversion of the semi-implicit operator P_{SI} . The latter is a parabolic operator, amenable to multilevel techniques, as described in [11].

4 Adaptive Mesh Refinement

The previous discussion has considered the generalization of the physics-based preconditioner proposed in Ref. [11] to the application at hand, without regard to the specifics of the spatial discretization employed. In what follows, we describe the AMR-specific details of our treatment of the MHD equations, with particular emphasis on 1) the spatial discretization at coarse-fine interfaces, 2) the generalization of multilevel solvers for SAMR grids, and 3) regriding and its impact on time integration.

4.1 Structured AMR Grids

Let $\Omega = [x_{lo}, y_{lo}] \times [x_{hi}, y_{hi}]$ be a rectangular computational domain. We create a discrete computational domain by subdividing $[x_{lo}, x_{hi}]$ into n_x subintervals with centers $x_i = x_0 + (i + \frac{1}{2})h_x$ with $h_x = (x_{hi} - x_{lo})/n_x$ for $i = 0, \dots, n_x - 1$. Each subinterval has faces located at $x_{i-\frac{1}{2}} = x_i - h_x/2$ and $x_{i+\frac{1}{2}} = x_i + h_x/2$. Likewise $[y_{lo}, y_{hi}]$ is partitioned into n_y subintervals with centers $y_j = y_0 + (j + \frac{1}{2})h_y$ with $h_y = (y_{hi} - y_{lo})/n_y$ for $j = 0, \dots, n_y - 1$ and faces $y_{j-\frac{1}{2}} = y_j - h_y/2$ and $y_{j+\frac{1}{2}} = y_j + h_y/2$. The tensor product of these subintervals partitions Ω into a collection of computational cells $\Omega^h = \{\Omega_{i,j}\}$ each with size $h_x \times h_y$ centered at coordinates (x_i, y_j) . These ideas are readily extended to the case where Ω is a union of non-overlapping rectangular regions, and we continue to use the same notation Ω^h to denote such a collection of computational cells. Such *regular grids* are in widespread use in computational science and engineering, and a great deal of high quality software that is tuned to regular grids, such as geometric multigrid methods, is available.

Let $K \geq 1$ and $\Omega_1 \equiv \Omega \supset \Omega_2 \supset \dots \supset \Omega_K$ be a nested set of subdomains of the computational domain Ω . For simplicity, assume that each Ω_ℓ , $2 \leq \ell \leq K$ is

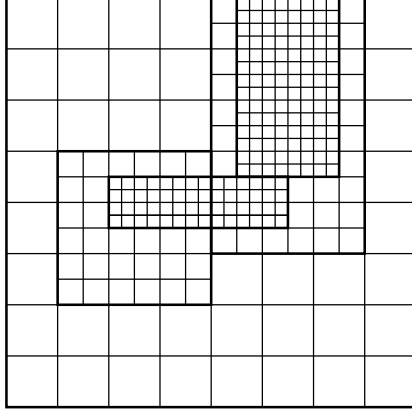


Fig. 1. Example of a multilevel SAMR grid with three levels.

a union of non-overlapping rectangular regions; these are the subregions of Ω where additional resolution is desired. A composite structured AMR (SAMR) grid Ω^c on Ω is a nested hierarchy of grids $\Omega_1^{h_1} \supset \Omega_2^{h_2} \supset \dots \supset \Omega_K^{h_K}$ consisting of K levels, with mesh spacing $h_1 > h_2 > \dots > h_K$, with the coarsest grid $\Omega_1^{h_1}$ covering Ω . Each level $\Omega_\ell^{h_\ell}$ consists of a union of non-overlapping rectangular regions, or *patches*, at the same resolution h_ℓ . When there is no risk of confusion we will drop the ℓ subscript and simply refer to Ω^{h_ℓ} . This hierarchical representation allows operations on Ω^c to be implemented as operations on individual levels Ω^{h_ℓ} , which in turn are decomposed into operations on individual patches, each of which covers a rectangular region $[x_{lo}, y_{lo}] \times [x_{hi}, y_{hi}]$. This property facilitates reuse of software written for regular grids. Figure 1 shows a SAMR grid with $K = 3$ and two patches on each of the two refinement levels. Note that while each level is nested in the next coarser level, there is no requirement that a patch at one refinement level is nested fully in a patch at another refinement level, i.e., a fine patch at refinement level l may lie over one or more coarser patches at refinement level $(l - 1)$. Figure 2 shows the decomposition of a fairly complex SAMR grid with six refinement levels into its constituent refinement levels and patches, as is encountered in our simulations.

4.2 Function evaluation on SAMR grids

For a viable JFNK solver, a given application only needs to provide methods to evaluate F , set up a preconditioner, and apply the preconditioner. By generalizing these steps to a SAMR grid hierarchy, JFNK can be readily adapted to SAMR applications. In this work, we use the PETSc parallel implementation of JFNK [3], which is made SAMR-aware via the PETSc-SAMRAI interfaces described in [39]. Considerations for evaluating F are described next.

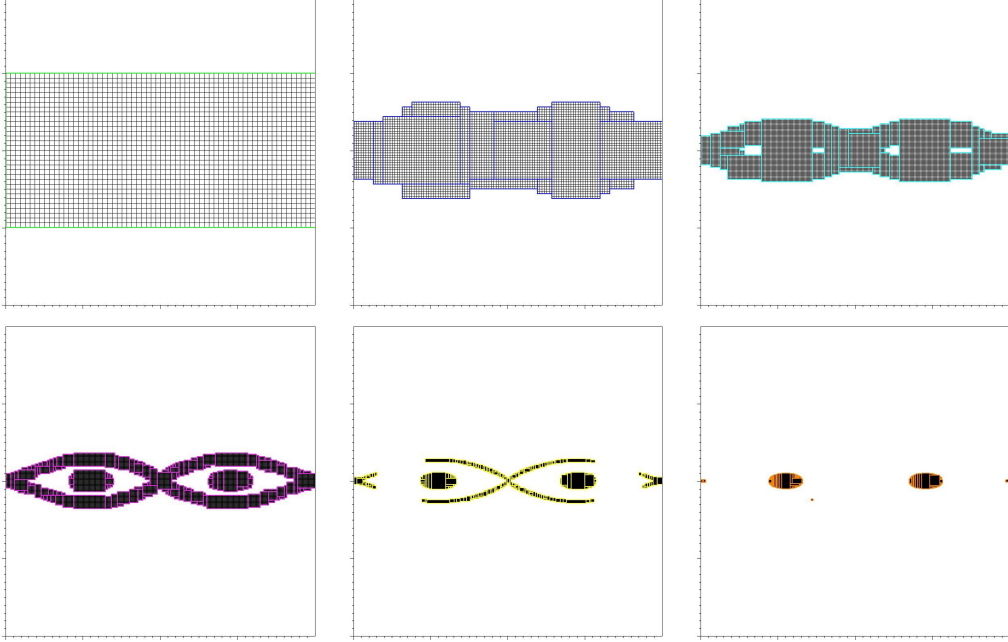


Fig. 2. The decomposition of a SAMR grid into its constituent refinement levels

4.2.1 Single grid discretization

In order to discretize F [given by (6)] in space, a cell-centered collocated finite-volume scheme is used for ω , J , Φ , and Ψ . The magnetic field, $\vec{B} = (B^1, B^2)^\top$, and the velocity, $\vec{v} = (u^1, u^2)^\top$, are also stored at cell centers, and are computed using centered differences from Ψ and Φ , respectively, using the discrete curl operations:

$$\begin{aligned} B_{i,j}^1 &= -\frac{\Psi_{i,j+1} - \Psi_{i,j-1}}{2h_y}, \\ B_{i,j}^2 &= \frac{\Psi_{i+1,j} - \Psi_{i-1,j}}{2h_x}, \end{aligned} \tag{12}$$

and

$$\begin{aligned} u_{i,j}^1 &= -\frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2h_y}, \\ u_{i,j}^2 &= \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2h_x}. \end{aligned} \tag{13}$$

We note that this discretization ensures that the divergence-free conditions on \vec{B} and \vec{v} are satisfied locally to numerical round-off.

Diffusive operators are discretized in each cell (i, j) by first computing approximate face-centered diffusive fluxes and then summing over the faces of each

cell resulting in the standard five-point finite-volume discretization:

$$\begin{aligned} \frac{1}{h_x h_y} \int_{\Omega_{i,j}} \nabla \cdot \nabla \Psi \, dA \approx & \frac{\Psi_{i+1,j} - \Psi_{i,j}}{h_x^2} - \frac{\Psi_{i,j} - \Psi_{i-1,j}}{h_x^2} + \\ & \frac{\Psi_{i,j+1} - \Psi_{i,j}}{h_y^2} - \frac{\Psi_{i,j} - \Psi_{i,j-1}}{h_y^2}. \end{aligned} \quad (14)$$

Advective quantities are discretized using standard centered differences. The standard cell-centered difference discretization for the gradient operator can be derived by using a variant of the Gauss theorem for gradients on each cell:

$$\int_{\Omega_{i,j}} \nabla \omega \, dA = \int_{\partial\Omega_{i,j}} \omega \mathbf{n} \, ds$$

where \mathbf{n} is the unit outward-facing normal on $\partial\Omega_{i,j}$, and approximating the face centered values by averaging from cell centers. This results in

$$\frac{1}{h_x h_y} \int_{\Omega_{i,j}} \nabla \omega \, dA = \frac{\omega_{i+1,j} - \omega_{i-1,j}}{2h_x} \vec{i} + \frac{\omega_{i,j+1} - \omega_{i,j-1}}{2h_y} \vec{j} \quad (15)$$

When applied to discretize quantities such as $\vec{B} \cdot \nabla \omega$ using a finite-volume formulation, we note that there is an implicit assumption that \vec{B} is constant over each cell.

4.2.2 Extension to SAMR grids

The discretizations described in the previous subsection are valid in the interiors of individual patches as well as the boundaries between two patches on the same refinement level. However, in order to maintain accuracy, changes are required at the boundaries between coarse and fine patches. Figure 3 (left) shows the interface between a coarse and fine patch. We use ghost cells (both coarse and fine) for communication at coarse-fine interfaces, as well as between patches in the same refinement level. A fine ghost-cell (Figure 3, center) overlaps one coarse cell. A coarse ghost-cell (Figure 3, right), however, lies underneath four fine cells when a refinement ratio of 2 is used.

For computations of fine-ghost-cell values at coarse-fine interfaces, data is quadratically interpolated from a combination of coarse and fine-grid cell data. Figure 4(left) shows the coarse grid cells that would be involved in performing quadratic tangential interpolation of coarse-grid data to align it with fine-grid data. Figure 4(right) shows the piecewise quadratic normal interpolation

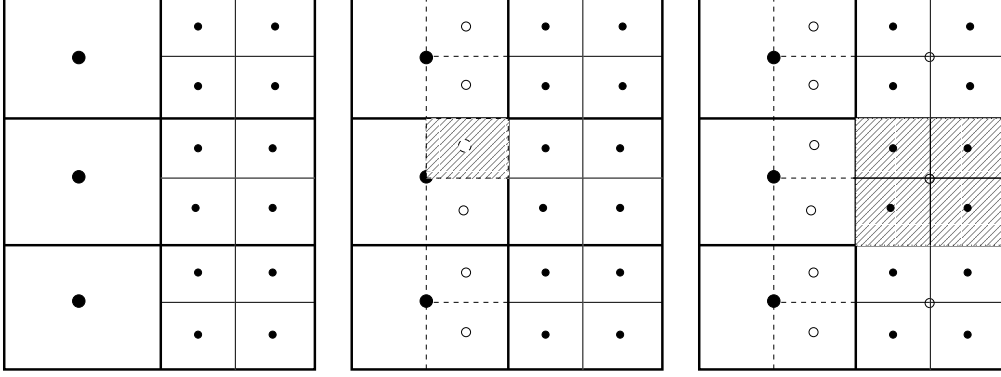


Fig. 3. (left) A coarse-fine interface. (center) Fine ghost cell. (right) Coarse ghost cell.

involving fine cells to calculate the fine-ghost-cell value. Once data has been interpolated to fine ghost-cells, fine cells at coarse-fine interfaces can be treated identically to cells that lie in the interior of the fine patch. Figure 4 only shows the simplest case where a sufficient number of coarse cells (in this case, three), are available to do standard quadratic interpolation tangential to the interface. In general, for block-structured AMR, many special cases need to be accounted for, where two or more fine patches may be adjacent to each other, resulting in very irregular coarse-fine interfaces (see for example Figure 2). We do not detail the adjustments needed in each of these cases to interpolate data quadratically, due to space limitations. However, [2] provides a glimpse of the types of adjustments required.

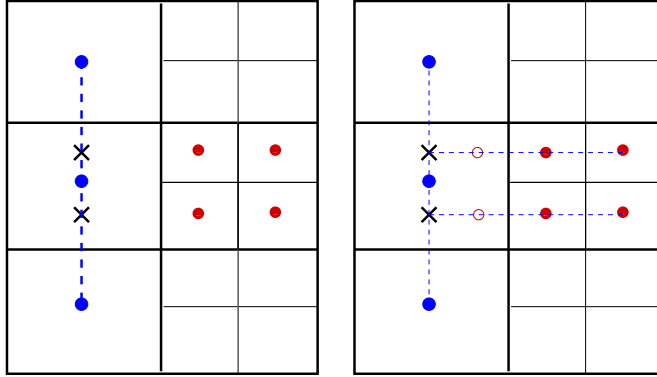


Fig. 4. Schematic of interpolation. (Left) Crosses show coarse grid data aligned with fine grid data by interpolation. (Right) Open circles denote fine ghost-cell data obtained by interpolation from aligned coarse data and fine grid data.

Regarding coarse ghost-cells at coarse-fine interfaces, their treatment is done as follows:

Magnetic and velocity fields: \vec{B} and \vec{v} are computed from Ψ and Φ using centered differences. While this maintains the vector fields divergence-free everywhere, it requires values of Ψ and Φ to be available at the coarse ghost-cells. A simple averaging of fine Ψ and Φ data to coarse ghost-cells produces

second-order accurate coarse values, and therefore the resulting \vec{B} and \vec{v} are only first-order accurate. Local first-order accuracy at coarse-fine interfaces is unlikely to affect the global second-order accuracy of the calculation when the number of coarse fine cells is small relative to the total number of cells. However, Ref. [1] gives examples of cases not unlike the situations present in our simulations where the number of coarse-fine interface cells can be up to 30% of the total number of cells. In such cases, global accuracy will be affected by the local first-order accuracy at coarse-fine interfaces. To avoid this situation, we use piecewise cubic interpolation of fine-grid data for Ψ , Φ to coarse ghost-cells. This, in turn, results in second-order discretizations for the vectors at coarse cells adjacent to the coarse-fine interface.

Diffusion operators: To compute the numerical diffusive fluxes at coarse cells adjacent to the coarse-fine interface, it is possible to use coarse ghost-cell values underlying the fine grid. However, for flux conservation, we compute the diffusive fluxes at fine cell faces (as described in Sec. 4.2.1) and average them down to provide the coarse flux at the coarse face.

Advection operators: Centered differences in each coarse cell are computed by first computing face-averaged values of the advected quantity as described in section 4.2.1. At coarse-fine interfaces, we follow Ref. [24] and find a coarse face-averaged value by averaging fine face-averaged data. (The alternative is to compute a coarse ghost-cell value directly from coarse data, but this does not work as well.)

4.3 *Regridding*

For fully dynamic AMR simulations, the grid hierarchy will change during the simulation as the solution evolves in time and space. This involves the addition of fine patches in regions where additional resolution is required and the removal of patches in regions where coarser resolution is sufficient. The regridding process involves using some refinement indicator to determine required grid resolution, and then constructing a new grid hierarchy based on this information. The ideal refinement indicator is a sharp estimate of the spatial error that is inexpensive to compute. When such an estimate is not available, refinement indicators that detect features in the solution, such as regions of large gradients or curvature, are used. We detail the refinement indicators we employ in this study in Sec. 5.

Once regridding is done, data needs to be transferred from the old grid hierarchy to the new one. Typically, the solution obtained on the old grid hierarchy is no longer a solution on the new grid hierarchy. This has been observed in our simulations as well as reported by others [6]. Two approaches are com-

mon in the literature. One approach, referred to as the "warm restart" [5], continues the time integration on the new grid hierarchy (with a time step comparable to that before regridding) using the interpolated data on the new grid hierarchy. The second approach, called a "cold restart" [5], alters the time step and possibly the time integration scheme to account for regridding and introduction of spatial errors.

In this study, we use a third approach. After interpolation of the required vectors (including current and previous time-step solutions required for the time integration scheme), we solve for the current time step solution on the new grid hierarchy using the solution interpolated from the old grid hierarchy as an initial guess. This synchronizes the current time step solution with the new grid hierarchy before advancing in time. Numerically, this procedure is robust, and avoids propagation of interpolation errors during regridding steps. A detailed evaluation of all three approaches is reserved for future work.

4.4 Preconditioning and the Fast Adaptive Composite Grid Method

Physics-based preconditioning, as described in section 3.2, requires the inversion of the parabolic operator $P_{SI} = \mathcal{L}_\eta - \theta^2 \nabla \cdot (\mathbf{B}_0 \mathcal{D}_\nu^{-1} \mathbf{B}_0^\top \nabla)$. The operator $\nabla \cdot (\mathbf{B}_0 \mathcal{D}_\nu^{-1} \mathbf{B}_0^\top \nabla)$ is negative definite and self-adjoint. The discretization of this operator follows [50] so that it is compact (i.e., on a nine-point stencil), and the resulting matrix is symmetric negative definite. We note that, at coarse-fine interfaces, symmetry is lost. The convective operator in P_{SI} is discretized using a first-order upwind scheme for robustness (instead of the centered differences employed to evaluate the nonlinear residual). This leads to an algebraically better conditioned operator that increases the robustness of the preconditioner but does not affect the accuracy of the solution, which is determined by (7).

On a SAMR grid, the inversion of P_{SI} is performed efficiently by using the Fast Adaptive Composite grid (FAC) method [35,36]. FAC extends techniques from multigrid on uniform grids to AMR grids. FAC solves problems on AMR grids by combining smoothing on refinement levels with a coarse-grid solve using an approximate solver, such as a V-cycle of multigrid. First we introduce some notation to describe the FAC algorithm:

- $I_c^\ell : \Omega^c \rightarrow \Omega_\ell^h$ and $I_\ell^c : \Omega_\ell^h \rightarrow \Omega^c$ respectively denote restriction and interpolation operators between the composite SAMR grid, Ω^c and an individual refinement level. Here we use bilinear interpolation for I_ℓ^c and simple averaging for I_c^ℓ .
- $I_{\ell+1}^\ell : \Omega_\ell^h \rightarrow \Omega_{\ell+1}^h$ and $I_{\ell+1}^{\ell+1} : \Omega_{\ell+1}^h \rightarrow \Omega_\ell^h$ respectively denote restriction and interpolation operators between consecutive refinement levels.

Algorithm 1. FAC

```

Initialize:  $r^c = f^c - \mathcal{L}^c u^c$ ;  $f^\ell = I_c^\ell r^c$ 
foreach  $\Omega_\ell^h$ ,  $\ell = J, \dots, 2$ 
  Smooth:  $\mathcal{L}^\ell e^\ell = f^\ell$ 
  Correct:  $u^c = u^c + I_\ell^c e^\ell$ 
  Update:  $r^c = f^c - \mathcal{L}^c u^c$ 
  Set      :  $f^{\ell-1} = I_c^{\ell-1} r^c$ 
Solve   :  $\mathcal{L}^1 e^1 = f^1$ 
Correct:  $u^c = u^c + I_1^c e^1$ 
foreach  $\Omega_\ell^h$ ,  $\ell = 2, \dots, J$ 
  Update:  $r^c = f^c - \mathcal{L}^c u^c$ 
  Set      :  $f^\ell = I_c^\ell r^c$ 
  Smooth:  $\mathcal{L}^\ell e^\ell = f^\ell$ 
  Correct:  $u^c = u^c + I_\ell^c e^\ell$ 

```

- \mathcal{L}^c is the composite fine grid discrete operator obtained by discretizing the PDE on Ω^c , and \mathcal{L}^ℓ approximates \mathcal{L}^c on level ℓ .

With this notation we can specify the FAC Method as in Algorithm 1. After an initial residual is computed, smoothing is done on each level to determine a correction to the solution on that level. The levels are treated sequentially, from finest to coarsest, followed by a solve on the coarsest grid and then smoothing and correction from the coarsest to the finest levels. Algorithm 1 depicts an FAC V-cycle; as with multigrid methods, it is possible to specify alternative schedules for visiting levels, such as slash cycles or W-cycles.

Algorithm 1 makes clear the *multiplicative* nature of FAC: the residual is updated with the latest correction information before each smoothing pass can proceed. To be fully effective, each smoothing pass must properly account for the data dependencies among different patches within a refinement level. In our calculations, we use red-black Gauss-Seidel smoothing on each refinement level; we also have the capability to use weighed-point-Jacobi or zebra-line Gauss-Seidel smoothing. The correction steps require synchronization of the composite grid solution to make it consistent on all refinement levels. Note that the residual update can, in principle, be computed only on the most recently corrected refinement level plus a small border on the next coarser level, but we have found that residual evaluation is not expensive enough to justify this optimization. On the coarsest level, we use one V-cycle of `hypr`'s [17] implementation of semi-coarsening multigrid [49].

5 Numerical results

This section introduces several challenging test cases with the goal of demonstrating two main aspects of our implicit AMR implementation: algorithmic performance, and its accuracy properties. In regards to performance, we demonstrate that the convergence properties of the iterative approach are essentially independent of the number of grid levels present in the simulation (for equivalent fine-level resolution), and that it has good scaling properties with respect to the total number of unknowns. Both these aspects are central in a scalable implicit AMR algorithm.

Accuracy and error propagation in the SAMR context is also a major subject of this section. Clearly, the main motivation for using an AMR-based strategy is to minimize the number of unknowns required to achieve a given error level, by using fine resolution only where it is needed. However, time integration on SAMR grids can pose additional difficulties and introduce sources of error not encountered in uniform-grid calculations, especially at the interface between coarse and fine refinement levels. For nonlinear problems, these errors can exhibit themselves in unexpected ways that are often hard to identify and offset. In fact, the potential exists that errors generated at coarse-fine interfaces, combined with the wrong temporal integrator, may overwhelm the simulation and result in errors that scale with the coarsest (instead of the finest) resolution in the computational mesh.

A fundamental aspect for optimal error control in SAMR is the adequate placement of patches within the domain. Careless placement of patches can in fact offset any gains that may be obtained with the additional resolution provided. Optimal placement of patches, in turn, requires suitable error estimators. We will demonstrate the importance of this issue numerically later in this section. For our test problems, we employ refinement indicators based on the magnitude of the current, J , and the curvature in the vorticity, ω , as a guide for patch placement. In particular, we compute the cell quantities

$$\tau_{i,j}^1 = \frac{|(J)_{i,j}|}{\max_{(i,j)} |(J)_{i,j}|}, \quad \tau_{i,j}^2 = \frac{|h_x^2(\omega_{xx})_{i,j}| + |h_y^2(\omega_{yy})_{i,j}|}{0.2 \max_{i,j} |\omega_{i,j}|}. \quad (16)$$

Here, ω_{xx} and ω_{yy} denote second-order partial derivatives of the vorticity in x and y , respectively. The maximum values for J and ω in (16) are calculated over each refinement level. Cells where $\tau_{i,j}^1 > \epsilon_1$ or $\tau_{i,j}^2 > \epsilon_2$ where ϵ_1 and ϵ_2 are user chosen thresholds, are tagged for refinement. In this application $\epsilon_1 = 0.65$ and $\epsilon_2 = 0.3$. Similar error estimators are used in Refs. [6] and [1].

For the accuracy tests, since analytic solutions are not available, the numerical error is computed by comparing a given SAMR simulation against a uniform

fine-grid solution (512×512 unless otherwise specified), obtained with an extremely small timestep.

In what follows, we discuss these issues for three test problems: a tearing mode problem, the island coalescence problem, and the tilt instability problem. All these problems feature the dynamic development of thin current layers, and benefit from an AMR treatment.

5.1 Tearing Mode Problem

The first problem we consider is the tearing mode problem of [11]. Tearing modes are resistive instabilities, with behavior strongly dependent on the details of the resistive layer at the rational surface (defined as the surface where $\vec{B} \cdot \vec{k} = 0$, with \vec{k} the wavevector of the magnetic perturbation). As the resistive layer thickness scales as $\sqrt{\eta}$, the resolution issues become increasingly challenging with smaller η .

Following [11], we pose the problem in a square domain $\Omega = [0, 4] \times [0, 1]$, with periodic boundary conditions in x and homogenous Dirichlet boundary conditions in y for all variables. The initial conditions for this problem are $\omega_0(x, y) = \Phi_0(x, y) = 0$, and Ψ_0 given by the Harris sheet equilibrium:

$$\Psi_0(x, y) = \frac{1}{\lambda} \ln[\cosh(\lambda(y - \frac{1}{2}))].$$

The initial current J_0 is found as $J_0 = \nabla^2 \Psi_0$. For the runs below, we have used $\lambda = 5$ and $\eta = \nu = 10^{-3}$.

Figure 5 shows the evolution of the system at different times on a dynamic SAMR grid with four refinement levels. The coarsest level is a uniform 32×32 grid, with the finest level providing the same resolution as a 256×256 uniform grid. The refinement levels track the evolution of the tearing mode, providing resolution only in localized regions.

5.1.1 Performance

Performance data for the tearing mode simulation are presented in Table 1. These results have been obtained using $\eta_k = 0.1$, $\epsilon_{rel} = 0.0$ and $\epsilon_{abs} = 1.0e-4$, and two iterations of the SI preconditioner. Within the preconditioner, two V(3,3) cycles of FAC were used to invert the semi-implicit operator outlined in Sec. 3.2. The implicit time step has been fixed to a multiple of the explicit CFL, $\Delta t = 140\Delta t_{CFL}$ (which corresponds to a time step of $\Delta t = 5$ for the

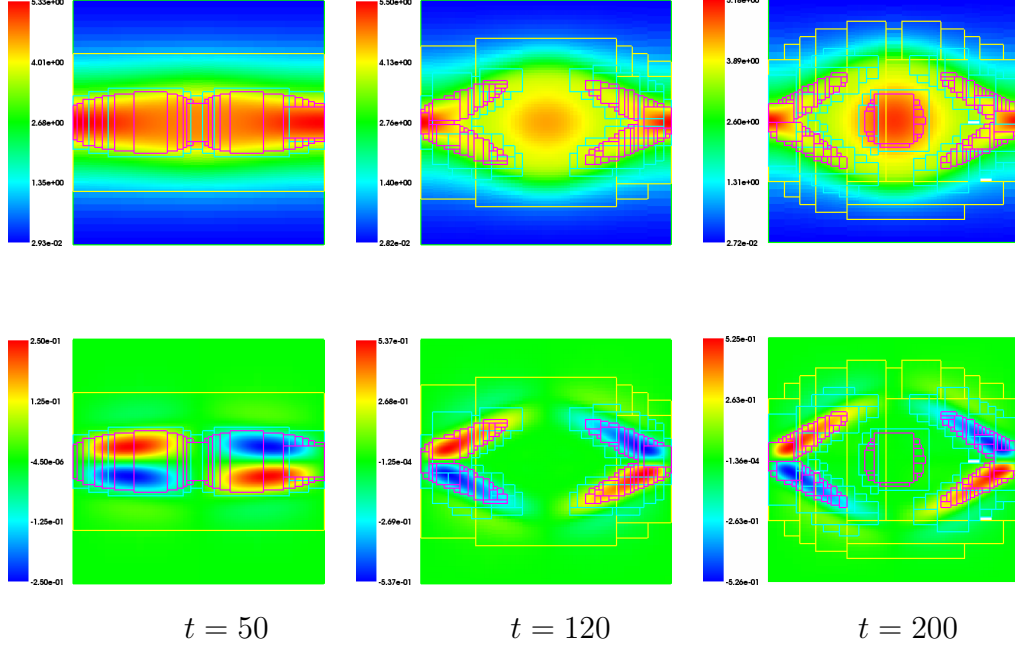


Fig. 5. Time snapshots of current (top) and vorticity (bottom) for the tearing mode problem.

coarsest 64×32 grid), and we have averaged the performance results over the course of the simulation up to $T_{max} = 250$.

Table 1

Summary of performance for tearing mode. *NNI*: average number of nonlinear iterations; *NLI*: average number of linear iterations.

	NNI					NLI				
Levels	1	2	3	4	5	1	2	3	4	5
64×32	2.4	2.7	2.8	2.8	2.7	8.2	10.3	14.1	15.8	16.4
128×64	2.3	2.6	2.7	2.6	—	8.0	12.9	15.1	16.1	—
256×128	2.0	2.1	2.4	—	—	9.4	13.9	12.8	—	—
512×256	2.0	2.1	—	—	—	12.4	13.3	—	—	—

The first column of the Table specifies the resolution of the coarse grid, while levels refers to the number of refinement levels active on the SAMR grid. Moving diagonally from left to right, for example from the cell corresponding to the 512×256 grid with one refinement level to the cell with a 64×32 base grid and five levels of refinement, we see that the number of nonlinear iterations, *NNI*, remains more or less constant. The number of linear iterations, *NLI*, grows slightly with resolution, both in single-grid and AMR computations. Nevertheless, it is apparent that we are obtaining very similar solver performance on a SAMR grid to what we would expect to see on a uniform grid with comparable fine resolution.

This substantially equivalent solver performance is obtained for significantly less computational effort and memory storage. For example, the 512×256 run requires a total wall-clock time of 29919 seconds while the equivalent 64×32 run with 4 levels of refinement takes 4998 seconds of wall-clock time. Furthermore, on average, the 64×32 run with 4 levels of refinement requires only 14% of the number of degrees of freedom that a uniform 512×256 run would require. The timing runs were obtained on a MacBook Pro with a 2.33 GHz Intel Core 2 Duo processor, 2 GB of RAM, running Mac OS X 10.4.10.

Similar results both in solver performance as well as time and memory savings were obtained in all our simulations.

5.1.2 Time-integration errors

We seek to characterize the importance of adequate interpolation order during regridding to avoid discretization errors, and of robust damping in the implicit temporal integration scheme to avoid error propagation throughout the domain.

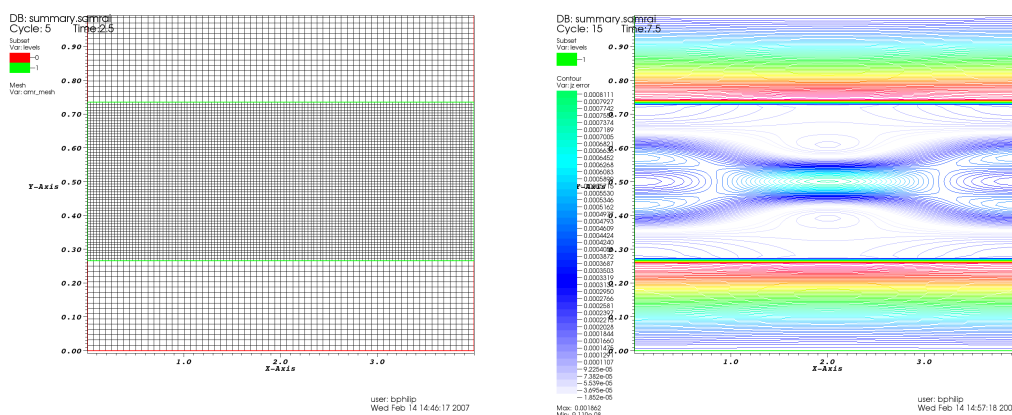


Fig. 6. Grid hierarchy and error in J after one regrid operation at $t = 2.5$ and several time steps up to $t = 7.5$.

A common approach for time integration on AMR grids is to use linear interpolation to transfer data from an old grid hierarchy to a new grid hierarchy during regridding. For problems with discontinuous coefficients and steep solution gradients [40], this works better than higher-order interpolation schemes. However, linear interpolation is not suitable for all problems. Figure 6 depicts the error in J on a SAMR grid after one regrid operation at $t = 2.5$ and subsequent time-stepping using Crank-Nicolson (CN) and quadratic interpolation at coarse-fine interfaces up to $t = 7.5$. Linear interpolation was used to transfer data from the old to the new grid hierarchy during regrid. The concentration of discretization errors at coarse-fine interfaces introduced during regridding is evident, despite the fact that quadratic interpolation was used at coarse-fine

interfaces during the time evolution. Furthermore, the poor damping properties of CN preserves the memory of generated errors, even if the location of the coarse-fine interfaces changes. To show this, we transfer the previous solution at $t = 7.5$ to a uniform mesh with resolution equivalent to the finest SAMR patch, and run the simulation further in time to $t = 42.5$. The result is depicted in Figure 7, and shows that the error in J generated during regridding at coarse-fine interfaces remains throughout the calculation. We note that the magnitude of the error does not appear to be changing significantly.

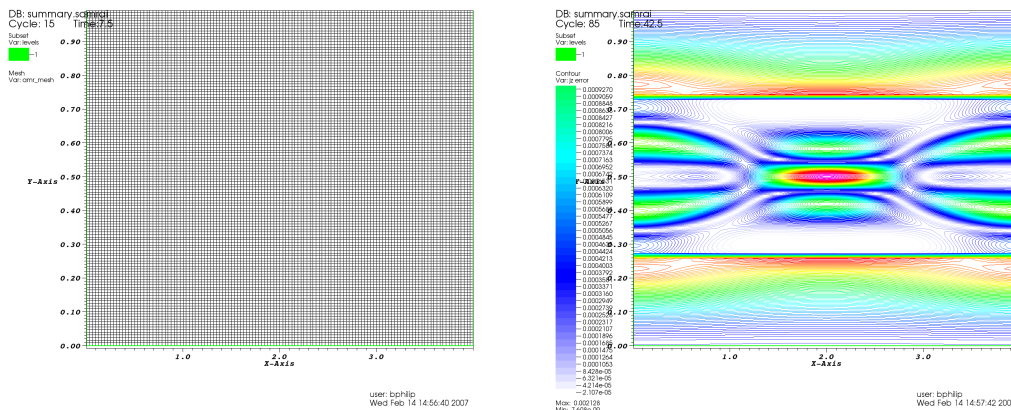


Fig. 7. Grid hierarchy and error in J after a second regrid operation to a uniform mesh at $t = 7.5$ and subsequent time-stepping until $t = 42.5$.

A related test with a different initial grid configuration is shown in Figure 8. In this case, the error in J is largest on the left and right coarse-fine interfaces at time $t = 7.5$. After several dynamical regridding operations, the solution has memory of errors generated at all coarse-fine interfaces during regridding, and at time $t = 42.5$ the error (Figure 9) traces all coarse-fine interfaces that have been present during the simulation. In this example, it is clear that, at some locations, the error is actually being amplified further, as for example at the left and right coarse-fine interfaces present in the initial grid hierarchy at $t = 7.5$.

The previous examples illustrate potential sources of error that can accumulate due to a combination of a poor temporal integrator (CN) and low-order interpolation during regridding, even if higher-order interpolation is used at coarse-fine interfaces. Low-order interpolation generates spatial errors that are not damped in time by the temporal scheme. Coarse-fine errors mostly disappear when sufficiently high-order interpolation is used (both at regrid operations and subsequently during time evolution) in combination with a robustly damping temporal scheme. Figure 10 shows plots of the error in J at times $t = 10$ and $t = 50$, using cubic interpolation at regrid, quadratic interpolation in between regrid operations, and BDF2 as the time integrator. In this case, we see that there is no accumulation of error at coarse-fine interfaces, even after several regrid operations have taken place.

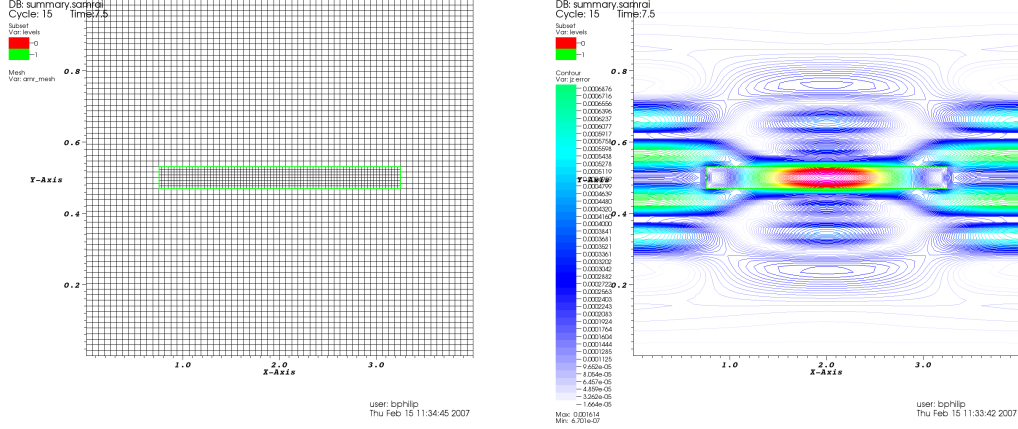


Fig. 8. Grid hierarchy and error in J at $t = 7.5$ after one regrid operation at $t = 2.5$.

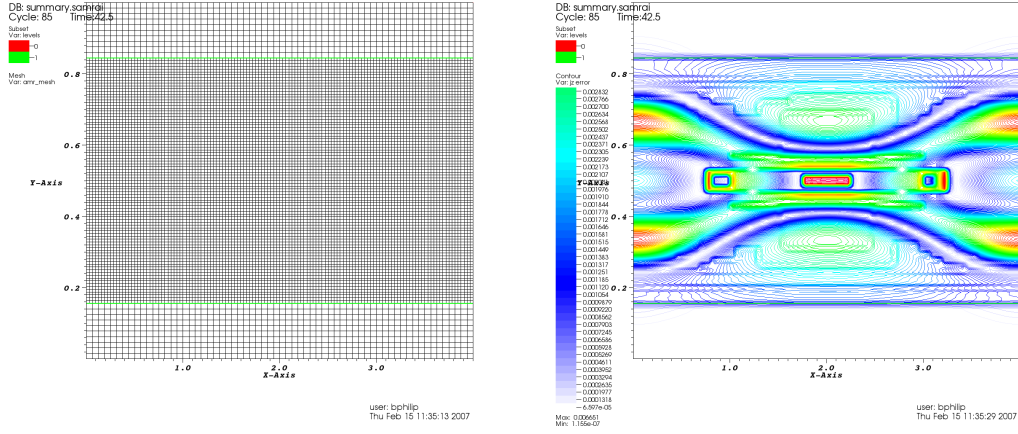


Fig. 9. Grid hierarchy and error in J at $t = 42.5$ after several regridding operations.

5.2 Island Coalescence

In the island coalescence problem, two magnetic islands (current channels) attract and reconnect. In resistive MHD, and during the reconnection process, a thin, elongated current sheet forms at the reconnection site, which governs the reconnection rate, and therefore the global dynamics [30]. As in the tearing mode problem, the current thickness scales as $\sqrt{\eta}$, and therefore the problem becomes computationally more challenging for smaller resistivities.

The island coalescence problem equilibrium is given by $\omega_0(x, y) = \Phi_0(x, y) = 0$, and Ψ_0 defined as [20]:

$$\Psi_0(x, y) = -\lambda_\Psi \ln \left[\cosh \left(\frac{y}{\lambda_\Psi} \right) + \epsilon \cos \left(\frac{x}{\lambda_\Psi} \right) \right], \quad (17)$$

where $\lambda_\Psi = \frac{1}{2\pi}$ is the current sheet equilibrium scale length, and $\epsilon = 0.2$ is the

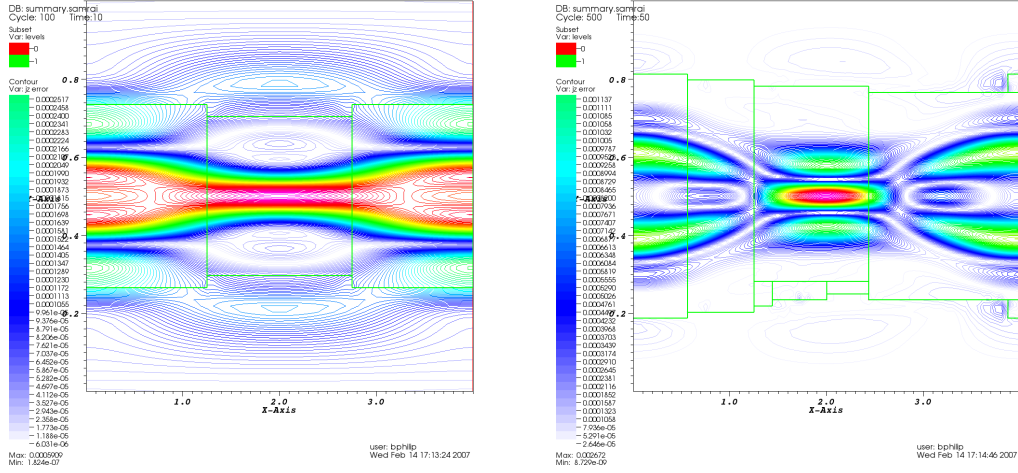


Fig. 10. Plots of error in J for $t = 10$ and $t = 50$ using BDF2 for time integration and cubic interpolation for regridding operations.

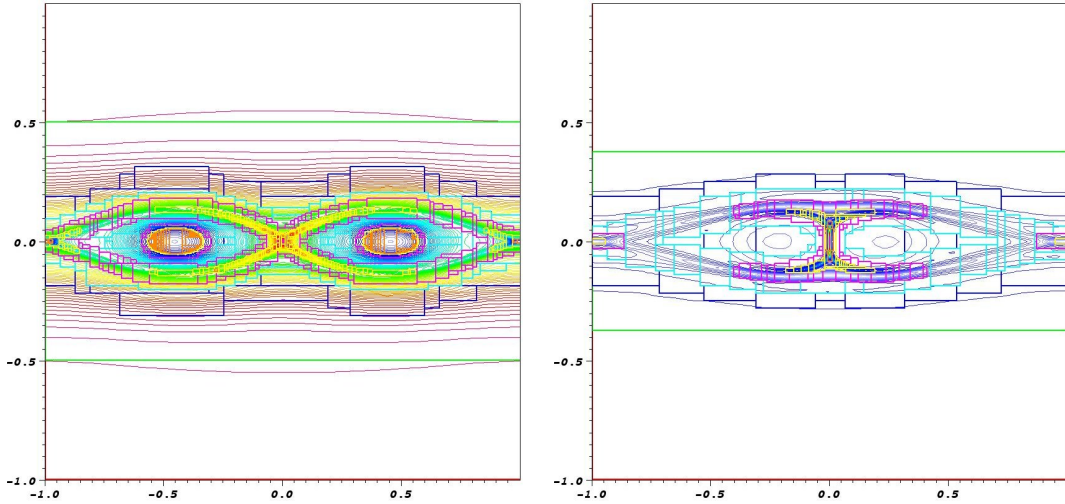


Fig. 11. Snapshots of the current at early stages of the coalescence process ($t = 4$, left) and at the peak of the reconnection ($t = 8$, right) during the coalescence process.

island width. The computational domain is $\Omega \in [-1, 1] \times [-1, 1]$. Boundary conditions are Dirichlet in y , and periodic in x . Both η and ν are set to 10^{-4} . The calculation is started with a perturbation in Ψ . Full dynamical SAMR regridding is employed to adapt to developing features. The island configuration at the early stages of the coalescence process is shown in Figure 11. The configuration at the peak of the reconnection rate, well in the nonlinear stage, is shown in Figure 11, and shows the formation of the current sheet in the symmetry plane between the islands. The multiscale nature of this problem is evident.

5.2.1 Numerical error generation and propagation

As mentioned earlier, the purpose of grid adaptation is to minimize the number of degrees of freedom required for a given simulation, while being compatible with a given numerical error level. In practice, the expectation is that the overall numerical error of the simulation does not increase due to the presence of patches, and the related coarse-fine boundary treatment. Otherwise, it would defeat the purpose of using a patch-based adaptive scheme in the first place.

We use the island coalescence problem to characterize the generation and temporal propagation of errors in our SAMR implementation. We focus on two main aspects of error generation in SAMR: 1) the treatment of coarse-fine interfaces at patches' boundaries, and 2) the placement of patches themselves. The former is fundamentally a spatial discretization issue, whereas the latter is more related to error estimation.

Figure 12 depicts several time histories of SAMR simulations featuring different base grids and levels of refinement. The error is obtained with an “exact” error detector, which compares the SAMR solution with the 512×512 reference solution. The point of this plot is to establish that the error is fundamentally a function of the finest resolution employed, and not a function of the base grid size or the number of levels of refinement. For instance, the 64b3l, 128b2l, and 256b1l simulations feature the *same* error (time histories are actually superimposed), while using different base grid refinements and number of grid levels. The same is true for 64b4l, 128b3l, and 256b2l (although error differences are more noticeable due to the log scale of the plot). Another side point from this figure is the impact that adding a level of refinement has on the overall error of the computation: the error decreases about 2 orders of magnitude on average.

The effect of choice of refinement indicator is depicted in Figure 13, where time histories of the numerical error resulting from two different error estimators are provided. The *exact* error estimator is compared against an *ad-hoc* indicator (16). The error plot labeled 128b3l-exact plots the magnitude of the $L2$ error on an AMR grid with 3 levels, when the placement of patches is determined by the “exact” error detector. Similarly, the error plot labeled 128b3l-feature plots the error on an AMR grid determined using (16). Clearly, the overall numerical error resulting from the *ad-hoc* approach is about two orders of magnitude larger, almost offsetting the accuracy gains obtained by adding refinement levels. This result underscores the importance of further research on reliable error estimators for adaptive-grid applications. This is left for future work.

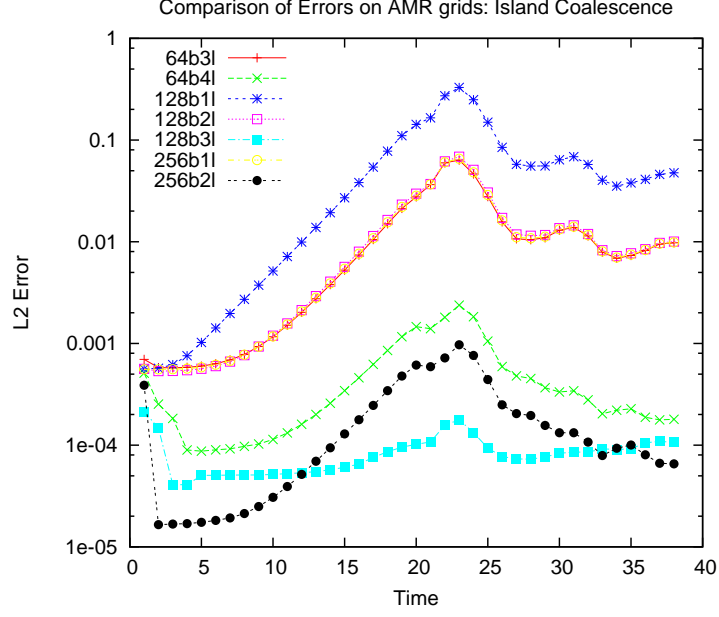


Fig. 12. Time histories of numerical errors for the island coalescence problem with various base grids and levels of refinement. The convention in the legend is $nbml$, which denotes an $n \times n$ base grid with m levels of refinement.

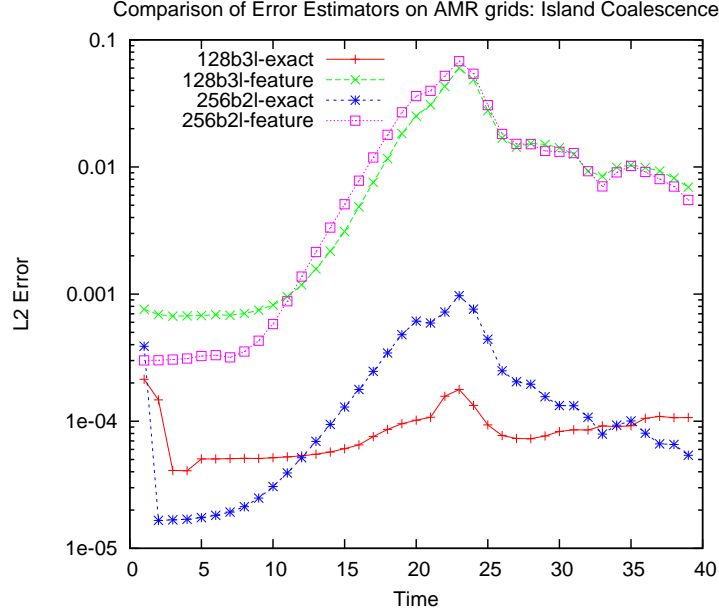


Fig. 13. Time histories of numerical errors for the island coalescence problem with different error estimators for patch placement.

5.3 Tilt Mode

The third model problem we consider is the tilt-mode instability [43,53,27]. The initial conditions are $\omega_0(x, y) = \Phi_0(x, y) = 0$, and Ψ_0 given by:

$$\Psi_0(x, y) = \begin{cases} \frac{2}{kJ_0(k)} J_1(kr) \cos(\theta) & \text{if } r \leq 1 \\ (r - \frac{1}{r}) \cos(\theta) & \text{if } r > 1 \end{cases}$$

with $J_0 = \nabla^2 \Psi_0$. The boundary conditions are periodic in x and Dirichlet in y for all variables. The system is perturbed from its initial equilibrium with a rotational perturbation in Φ of the form: $\delta\Phi = 10^{-3}e^{-r^2}$. The domain for this problem is $\Omega = [-2\pi, 2\pi] \times [-5, 5]$. The parameter k is the zero of the Bessel function of the first kind, i.e., $J_1(k) = 0$, and both η and ν are set to 10^{-3} . The refinement criteria for this problem is also given by (16). Figure 5.3 shows snapshots of the current and vorticity during the evolution of the tilt instability at times $t = 4.0$ (early in the linear phase) and $t = 7.0$ (well in the nonlinear regime). This calculation required seven levels of refinement starting from a coarse initial 64×64 mesh.

As evolving features in this problem are extremely small compared to the domain size, this AMR calculation on average only required 0.36% of the degrees of freedom of a uniform grid calculation (a uniform grid calculation would have required 67108864 degrees of freedom, which corresponds to a 4096×4096 uniform grid with 4 unknowns per grid cell). This example serves to illustrate the significant savings that AMR can provide over uniform grid calculations. We note that the relatively small number of degrees of freedom enabled us to perform this calculation on a workstation, while a parallel machine would have been required to perform the uniform-grid calculation.

6 Conclusions

We have described the implementation of a scalable, fully implicit, SAMR simulation tool for 2D reduced resistive MHD. The tool employs Jacobian-free Newton-Krylov methods as the solver engine. We use the reduced MHD solver developed in [11] as a starting point, albeit with several modifications. Following [51], we have reformulated the original problem (in terms of poloidal flux, vorticity, and streamfunction) using the current as a dynamic variable instead of the poloidal flux. This avoids issues with SAMR and the high-order differentiation of the latter in terms like $\vec{B} \cdot \nabla J$.

We have also extended the optimal “physics-based” approach developed in [11] (which employed multigrid solver technology in the preconditioner for scalability) for this application in two ways. Firstly, we have adapted the preconditioner formulation to deal with J instead of Ψ , as required. Secondly, we have extended the multilevel treatment in [11] to SAMR grids using the well-known Fast Adaptive Composite grid (FAC) method [35]. As a result, our approach inherits the algorithmic benefits of a multilevel treatment *and* the accuracy benefits of dynamic grid adaptation.

We have demonstrated such benefits with several challenging tests. A grid convergence study has shown that the solver performance is independent of

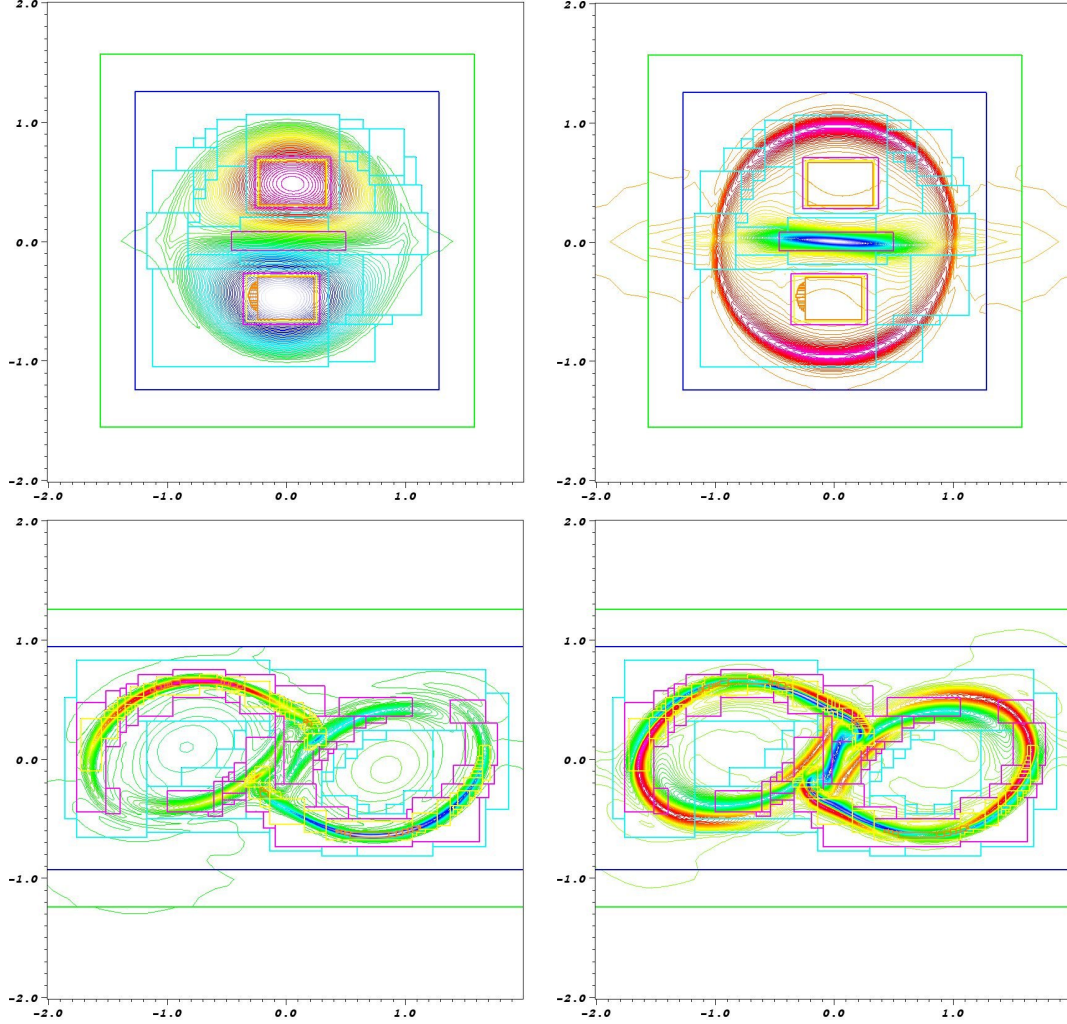


Fig. 14. Current (left) and vorticity (right) during the evolution of the tilt instability at times $t = 4.0$ (top) and $t = 7.0$ (bottom).

the number of grid levels and only depends of the finest resolution considered, and that it scales well with grid refinement. The study of error generation and propagation in our SAMR implementation demonstrates that piecewise cubic interpolation at coarse-fine interfaces, combined with a robustly damping second-order temporal scheme such as BDF2, is required to minimize impact of such interfaces on the overall error of the computation. We also demonstrate that our implementation features the desired property that the overall numerical error is dependent only on the finest resolution level considered, and not on the base-grid refinement or on the number of refinement levels present during the simulation.

An open aspect in our implementation is the use of more grounded, rigorous error estimators to select refinement patches, instead of our *ad-hoc* approach. In fact, we have demonstrated numerically that our *ad-hoc* error estimator, while effective, is far from optimal when compared with an *exact* one. In future

work, we will explore more rigorous error estimators, such as the τ (or two-grid) error estimator, which employs differences between two grid resolutions to estimate the truncation error. The approach, which is rigorous in the context of linear, conservative operators, can be used effectively for non-linear ones (see e.g. Refs. [7,19,34,18] for theory and practical implementation details and effectiveness of this error estimator in various contexts; in particular, Ref. [18] employs it for an anisotropic AMR implementation).

Acknowledgments

The authors acknowledge useful discussions with D. A. Knoll. The work was funded by the Los Alamos Directed Research and Development program at Los Alamos National Laboratory, operated for DOE under contract No. DE-AC52-06NA25396, and by the DOE Office of ASCR program in Applied Mathematical Sciences.

References

- [1] M. J. AFTOSMIS AND M.J. BERGER, *Multilevel error estimation and adaptive h-refinement for cartesian meshes with embedded boundaries*, in 40th AIAA Aerospace Sciences Meeting and Exhibit, no. 2002-0863 in AIAA Paper, January 2002.
- [2] A. S. ALMGREN, J. B. BELL, P. COLELLA, L. H. HOWELL, AND M. L. WELCOME, *A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations*, J. Comput. Phys., 142 (1998), pp. 1–46.
- [3] S. BALAY, W. D. GROPP, L. CURFMAN-McINNES, AND B. F. SMITH, *PETSc Users Manual*, Tech. Report ANL-95/11 - Revision 2.1.6, Argonne National Laboratory, 2004.
- [4] DINSHAW S. BALSARA, *Divergence-free adaptive mesh refinement for magnetohydrodynamics*, J. Comput. Phys., 174 (2001), pp. 614–648.
- [5] MARTIN BERZINS, PHILIP J. CAPON, AND PETER K. JIMACK, *On spatial adaptivity and interpolation when using the method of lines*, Applied Numerical Mathematics: Transactions of IMACS, 26 (1997), pp. 117–133.
- [6] J. G. BLOM AND J. G. VERWER, *Algorithm 759: Vlugs3: a vectorizable adaptive-grid solver for PDEs in 3d;part ii. code description*, ACM Trans. Math. Softw., 22 (1996), pp. 329–347.
- [7] A. BRANDT, *Multigrid techniques: 1984 guide, with applications to fluid dynamics*, gmd studien nr. 85, GMD, GMD-AIW, Postfach 1240, D-5205, St. Augustin 1, Germany, 1984. 191 pages.

- [8] P. N. BROWN AND Y. SAAD, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.
- [9] L. CHACÓN, *An optimal, parallel, fully implicit newton-krylov solver for three-dimensional visco-resistive magnetohydrodynamics*, submitted to Phys. Plasmas, (2007).
- [10] L. CHACÓN AND D. A. KNOLL, *A 2D high- β Hall MHD implicit nonlinear solver*, J. Comput. Phys., 188 (2003), pp. 573–592.
- [11] L. CHACÓN, D. A. KNOLL, AND J. M. FINN, *An implicit, nonlinear reduced resistive MHD solver*, J. Comput. Phys., 178 (2002), pp. 15–36.
- [12] J. CRANK AND P. NICOLSON, *A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type*, Proc. Camb. Phil. Soc., 43 (1947), pp. 50–67.
- [13] C. F. CURTISS AND J. O. HIRSCHFELDER, *Integration of stiff equations*, Proc. Nat. Acad. Sci., 38 (1952), pp. 235–243.
- [14] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
- [15] J. F. DRAKE AND T. M. ANTONSEN, *Nonlinear reduced fluid equations for toroidal plasmas*, Phys. Fluids, 27 (1984), pp. 898–908.
- [16] S. C. EISENSTAT AND H. F. WALKER, *Globally convergent inexact Newton methods*, SIAM J. Optimization, 4 (1994), pp. 393–422.
- [17] R. D. FALGOUT AND U. MEIER YANG, *hypre: a library of high performance preconditioners*, in Computational Science - CARS 2002 Part III, P. M. A. Sloot, C. J. K. Tan, J. J. Dongarra, and A. G. Hoekstra, eds., vol. 2331 of Lecture Notes in Computer Science, New York, 2002, Springer-Verlag, pp. 632–641.
- [18] L. FERM AND P. LOTSTEDT, *Anisotropic grid adaptation for Navier-Stokes equations.*, J. Comput. Phys., 190 (2003), pp. 22 – 41.
- [19] L. FERM AND P. L. LOTSTEDT, *Adaptive error control for steady state solutions of inviscid flow*, SIAM Journal on Scientific Computing, 23 (2002), pp. 1777 – 98.
- [20] J. M. FINN AND P.K. KAW, *Coalescence instability of magnetic islands*, Phys. Fluids, 20 (1977), pp. 72–78.
- [21] C. W. GEAR, *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, 1971.
- [22] K. GERMASCHEWSKI, A. BHATTACHARJEE, AND C.-S. NG, *The Magnetic Reconnection code: an AMR-based fully implicit simulation suite*, in Numerical Modeling of Space Plasma Flows, N. B. Pogorelov and G. P. Zank, eds., vol. 359 of ASP Conference Series, 2006.

- [23] T. I. GOMBOSI, *Solution-adaptive magnetohydrodynamics for space plasmas: Sun-to-earth simulations.*, Computing in science & engineering, 6 (2004), pp. 14 – 35.
- [24] D. M. GREAVES, *Simulation of viscous water column collapse using adapting hierarchical grids*, International Journal for Numerical Methods in Fluids, 50 (2006), pp. 693–711.
- [25] R. D. HAZELTINE, M. KOTSCHENREUTHER, AND P. J. MORRISON, *A four-field model for tokamak plasma dynamics*, Phys. Fluids, 28 (1985), pp. 2466–2477.
- [26] R. D. HORNUNG AND S. KOHN, *Managing application complexity in the SAMRAI object-oriented framework*, Concurrency Comput.: Pract. Exp., 14 (2002), pp. 347–368.
- [27] S. C. JARDIN AND J. A. BRESLAU, *Implicit solution of the four-field extended-magnetohydrodynamic equations using high-order high-continuity finite elements*, Physics of Plasmas, 12 (2005), p. 056101.
- [28] C. T. KELLEY, *Iterative methods for linear and nonlinear equations*, SIAM, Philadelphia, 1995.
- [29] R. KEPPENS, *Adaptive mesh refinement for conservative systems: Multi-dimensional efficiency evaluation.*, Computer Physics Communications, 153 (2003), pp. 317 – 339.
- [30] D. A. KNOLL AND L. CHACÓN, *Coalescence of magnetic islands, sloshing, and the pressure problem.*, Phys. Plasmas, 13 (2006), pp. 32307 – 1.
- [31] D. A. KNOLL AND D. E. KEYES, *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.
- [32] D. A. KNOLL AND P. R. MCHUGH, *Enhanced nonlinear iterative techniques applied to a nonequilibrium plasma flow*, SIAM J. Sci. Comput., 19 (1998), pp. 291–301.
- [33] S. LANKALAPALLI, J. E. FLAHERTY, M. S. SHEPHARD, AND H. STRAUSS, *An adaptive finite element method for magnetohydrodynamics*, Journal of Computational Physics, 225 (2007), pp. 363 – 381.
- [34] P. LOTSTEDT, S. SODERBERG, A. RAMAGE, AND L. HEMMINGSSON-FRANDE, *Implicit solution of hyperbolic equations with space-time adaptivity*, BIT, 42 (2002), pp. 134 – 58.
- [35] S. MCCORMICK, *Multilevel Adaptive Methods for Partial Differential Equations*, SIAM, Philadelphia, PA, 1989.
- [36] S. F. MCCORMICK AND J. W. THOMAS, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp., 46 (1986), pp. 439–456.

- [37] P. R. MCHUGH AND D. A. KNOLL, *Inexact Newton's method solution to the incompressible Navier-Stokes and energy equations using standard and matrix-free implementations*, AIAA J., 32 (1994), pp. 2394–2400.
- [38] C. S. NG, D. ROSENBERG, K. GERMASCHEWSKI, A. POUQUET, AND A. BHATTACHARJEE, *A comparison of spectral element and finite difference simulations with adaptive mesh refinement for the MHD island coalescence instability problem*, New journal of physics. in preparation.
- [39] M. PERNICE AND R. D. HORNING, *Newton-Krylov-FAC methods for problems discretized on locally refined grids*, Comput. Vis. Sci., 8 (2005), pp. 107–118.
- [40] MICHAEL PERNICE AND BOBBY PHILIP, *Solution of equilibrium radiation diffusion problems using implicit adaptive mesh refinement*, SIAM J. Sci. Comput., 27 (2006), pp. 1709–1726.
- [41] B. PHILIP, M. PERNICE, AND L. CHACÓN, *Solution of reduced resistive magnetohydrodynamics using implicit adaptive mesh refinement*, in Domain Decomposition Methods in Science and Engineering XVI, Olof B. Widlund and David E. Keyes, eds., vol. 55 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2007, pp. 725–731.
- [42] K. G. POWELL, *Parallel, AMR MHD for global space weather simulations*, Lecture notes in computational science and engineering, 41 (2005), pp. 473 – 490.
- [43] R. L. RICHARD, R. D. SYDORA, AND M. ASHOUR-ABDALLA, *Magnetic reconnection driven by current repulsion*, Physics of Fluids B: Plasma Physics, 2 (1990), pp. 488–494.
- [44] D. ROSENBERG, A. POUQUET, AND P. D. MININNI, *Adaptive mesh refinement with spectral accuracy for magnetohydrodynamics in two space dimensions*, New journal of physics, 9 (2007), pp. 304 –.
- [45] Y. SAAD AND M.H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
- [46] R. SAMTANEY, *Adaptive mesh refinement for MHD fusion applications*, Lecture notes in computational science and engineering, 41 (2005), pp. 491 – 503.
- [47] R. SAMTANEY, P. COLELLA, T. J. LIGOCKI, D. F. MARTIN, AND S. C. JARDIN, *An adaptive mesh semi-implicit conservative unsplit method for resistive MHD*, in SciDAC 2005. San Francisco, June 26-30. Journal of physics: Conference series., A. Mezzacappa et. al., ed., 2005.
- [48] R. SAMTANEY, S. C. JARDIN, P. COLELLA, AND D. F. MARTIN, *3D adaptive mesh refinement simulations of pellet injection in tokamaks*, Computer Physics Comm., 164 (2004), pp. 220–228.
- [49] S. SCHAFFER, *A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients*, SIAM J. Sci. Comput., 20 (1999), pp. 228–242.

- [50] MIKHAIL SHASHKOV AND STANLY STEINBERG, *Support-operator finite-difference algorithms for general elliptic problems*, J. Comput. Phys., 118 (1995), pp. 131–151.
- [51] H. STRAUSS AND D. LONGCOPE, *An adaptive finite element method for magnetohydrodynamics*, J. Comput. Phys., 147 (1998), pp. 318–336.
- [52] H. R. STRAUSS, *Nonlinear, 3-dimensional magnetohydrodynamics of noncircular tokamaks*, Phys. Fluids, 19 (1976), pp. 134–140.
- [53] H. R. STRAUSS AND D. W. LONGCOPE, *An adaptive finite element method for magnetohydrodynamics*, J. Comput. Phys., 147 (1998), pp. 318–336.
- [54] G. TÓTH, D. L. DE ZEEUW, T. I. GOMBOSI, AND K. G. POWELL, *A parallel explicit/implicit time stepping scheme on block-adaptive grids*, J. of Comput. Phys., 217 (2006), pp. 722 – 58.